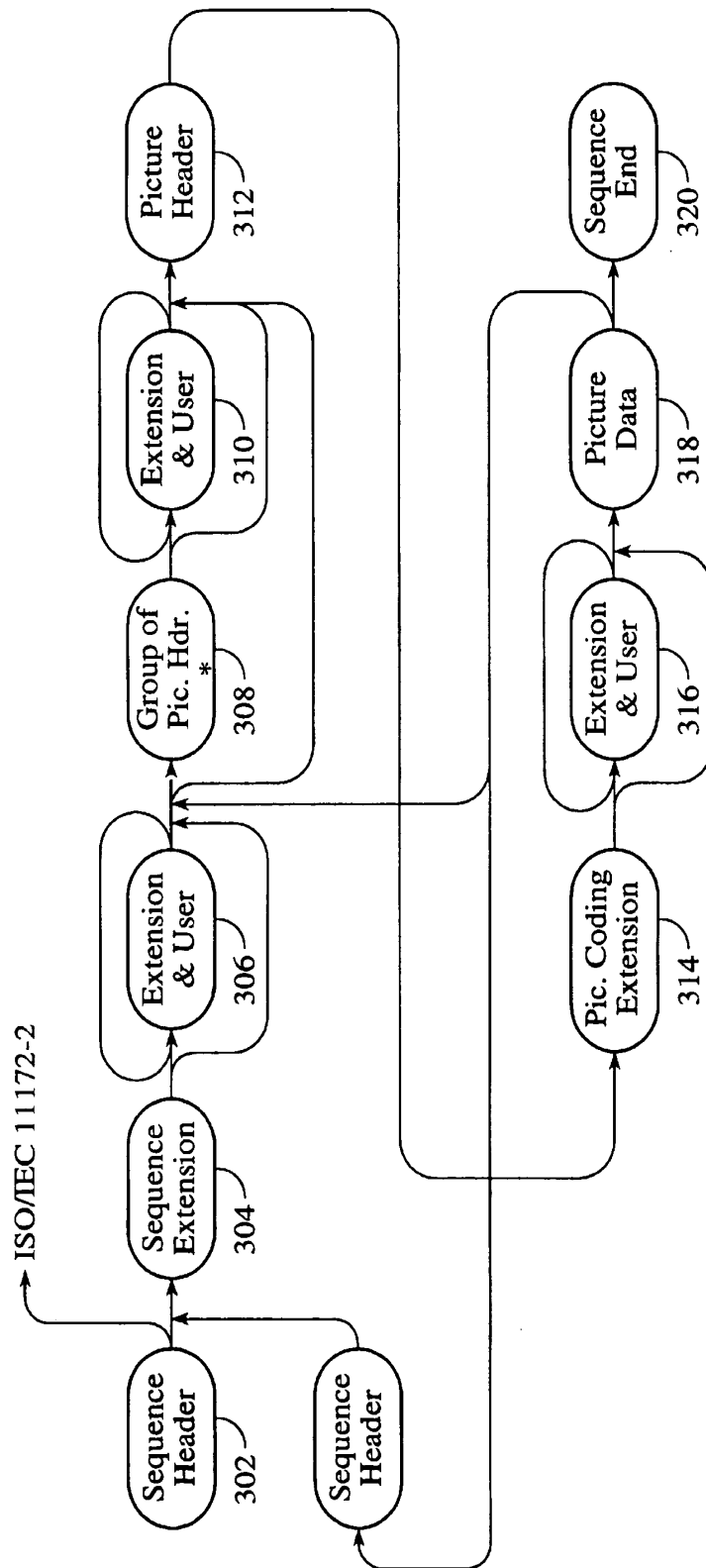


FIG. 2
(PRIOR ART)

EXHIBIT B



* After a GOP the first picture shall be an I-picture

FIG. 3A
(PRIOR ART)

video_sequence0 {	<u>300</u>	No. of bits	Mnemonic
next_start_code0			
sequence_header0	<u>302</u>		
if (nextbits0 = extension_start_code) {			
sequence_extension0	<u>304</u>		
do {			
extension_and_user_data(0)	<u>306</u>		
do {			
if (nextbits0 = group_start_code) {			
<u>308</u> group_of_pictures_header0			
<u>310</u> extension_and_user_data(1)			
}			
<u>312</u> picture_header0			
<u>314</u> picture_coding_extension0			
<u>316</u> extensions_and_user_data(2)			
<u>318</u> picture_data0			
} while ((nextbits0=picture_start_code)			
(nextbits0 = group_start_code))			
if (nextbits0! = sequence_end_code) {			
sequence_header0			
sequence_extension0			
}			
} while(nextbits0!=sequence_end_code)			
} else {			
/*ISO/IEC 11172-2 */			
}			
sequence_end_code	<u>320</u>	32	bslbf
}			

FIG. 3B
(PRIOR ART)

picture_header0 {	<u>312</u>	No. of bits	Mnemonic
picture_start_code	<u>502</u>	32	bslbf
temporal_reference	<u>504</u>	10	uimsbf
picture_coding_type		3	uimsbf
vbv_delay		16	uimsbf
if(picture_coding_type = 2 picture_coding_type = 3) {			
full_pel_forward_vector		1	
forward_f_code		3	uimsbf
}			
if (picture_coding_type = 3) {			
full_pel_backward_vector		1	
backward_f_code		3	uimsbf
}			
while (nextbits0 = '1') {			
extra_bit_picture /* with the value '1' */		1	uimsbf
extra_information_picture		8	
}			
extra_bit_picture /* with the value '0' */		1	uimsbf
next_start_code0			
}			

FIG. 5 (PRIOR ART)

group_of_pictures_header0 {	<u>308</u>	No. of bits	Mnemonic
group_start_code ~ 604		32	bslbf
time_code		25	bslbf
closed_gop ~ 602		1	uimsbf
broken_link		1	uimsbf
next_start_code0			
}			

FIG. 6 (PRIOR ART)

At encoder input <u>1602</u>	1	2	3	4	5	6	7	8	9	10	11	12	13
	I	B	B	P	B	B	P	B	B	I	B	B	P
At encoder output <u>1604</u>	1	4	2	3	7	5	6	10	8	9	13	11	12
	I	P	B	B	P	B	B	I	B	B	P	B	B
At decoder output <u>1606</u>	1	2	3	4	5	6	7	8	9	10	11	12	13
	I	B	B	P	B	B	P	B	B	I	B	B	P

FIG. 16